

# We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,500

Open access books available

136,000

International authors and editors

170M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index  
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?  
Contact [book.department@intechopen.com](mailto:book.department@intechopen.com)

Numbers displayed above are based on latest data collected.  
For more information visit [www.intechopen.com](http://www.intechopen.com)



# Introducing Machine Learning Models to Response Surface Methodologies

*Yang Zhang and Yue Wu*

## Abstract

Traditional response surface methodology (RSM) has utilized the ordinary least squared (OLS) technique to numerically estimate the coefficients for multiple influence factors to achieve the values of the responsive factor while considering the intersection and quadratic terms of the influencers if any. With the emergence and popularization of machine learning (ML), more competitive methods have been developed which can be adopted to complement or replace the traditional RSM method, i.e. the OLS with or without the polynomial terms. In this chapter, several commonly used regression models in the ML including the improved linear models (the least absolute shrinkage and selection operator model and the generalized linear model), the decision trees family (decision trees, random forests and gradient boosting trees), the model of the neural nets, (the multi-layer perceptrons) and the support vector machine will be introduced. Those ML models will provide a more flexible way to estimate the response surface function that is difficult to be represented by a polynomial as deployed in the traditional RSM. The advantage of the ML models in predicting precise response factor values is then demonstrated by implementation on an engineering case study. The case study has shown that the various choices of the ML models can reach a more satisfactory estimation for the responsive surface function in comparison to the RSM. The GDBT has exhibited to outperform the RSM with an accuracy improvement for 50% on unseen experimental data.

**Keywords:** response surface methodology (RSM), machine learning (ML), regression, the least absolute shrinkage and selection operator (LASSO), generalized linear model (GLM), decision trees, random forests, gradient boosting decision trees (GBDT), multiple-layer perceptrons (MLP), support vector regression (SVR)

## 1. Introduction

Response surface methodology (RSM) is an intersection of the experimental design, the objective optimization and the statistical modeling. RSM aims to identify the adequate mathematical model with the optimal selection of the influence factors to predict the responsive factor and to obtain the extremum of the model under the constrained numerical intervals or categorical levels of the influence factors. The objectives are achieved through designing and conducting a series of experiments to

collect the necessary amount of experimental data to approximate the mathematical model.

RSM was originally proposed and described in [1] and some other papers are subsequently published to contribute to the development of the RSM [2, 3]. Review papers began to appear starting from [4] that summarized the utilization of RSM in the chemical and processing fields. It is followed by a few more recent review papers in the application of the physical and science area [5, 6] and the biometric area [7] and some books in the related subjects as well [8, 9]. RSM has quickly gained popularity in empirical modeling in physical experiments to replace or complement the traditional presumed approach [10] where the theoretical knowledge of the experimental systems are available since its appearing. It is as well utilized in modeling the numerical experiments together with the simulation-based methods such as the finite element analysis in the application of the design optimization [11]. RSM contains three skeletal concepts including the estimation of the mathematical model or function, the design of experiments (DoE) and the validation and representation of the postulated mathematical function. Those three steps are likely to be insufficient to conduct only once and will require iteration in practice to achieve a satisfactory result [4].

The most commonly used postulation for the mathematical model is first-order or higher-order polynomials [12, 13]. Despite the wide implementation in chemistry and chemical engineering, it is inevitable that under certain circumstances, the polynomials are inadequate to approximate the underlying RMS functions (e.g. non-linear systems). Especially with the rapid development of information technology in the recent few decades, the utilization of RMS has been spreading to cover many other fields such as civil [14], advanced manufacturing [15, 16], and biomedical engineering [17, 18] and agricultural and food science [19, 20]. Experimental data has become much easier to collect, process and cache, parallel to which is the emergence of machine learning.

Machine learning (ML) is a process of the model building using experimental data or past experience in order to solve a given problem [21, 22]. It enhances the RMS by fitting a rather wide range of approximation models to achieve the responsive surface function. The whole process of ML model construction innately cooperates with the model estimation and model validation stages of the RSM [22]. Additionally, many ML models intrinsically select the most significant influence factors during the model construction process (e.g. the LASSO [23], the GLM and decision trees based models [24, 25]). This nature of the ML models will help to reduce the chance for attempting different DoE to identify the most appropriate influence factor combination and therefore contribute to diminishing the repetitiveness of the three-step cycle and reducing the total experimental runs and cost. Indeed, in cases where the experimental data is extremely expensive or difficult to obtain such as those in the biochemistry field [18, 19], the polynomial approximation and the corresponding DoE methods such as the full or partial factorial design [26], central composite design [27] and the Taguchi's experimental designs [26, 28] are still of utmost importance.

While the ML methods may not be suitable in certain scenarios where expensive time and financial cost are associated with the physical experiments, some techniques are still worth to be explored and utilized to replace or complement the traditional polynomial approach. The objective of the chapter is to introduce some of the linear and non-linear ML models to estimate the responsive surface function under the fair assumption of a reasonable cost and easiness in obtaining enough amount of experimental data. The book chapter is divided into 4 sections, the following section describing the frequently used ML models in detail, Section 3 implementing an engineering example to demonstrate the advantages of those ML

models in comparison to the traditional polynomial postulation and Section 4 summarizing the content in the chapter and discussing further research direction.

## **2. The machine learning approach**

### **2.1 The model construction, validation and testing**

Before diving into the various ML models to estimate the responsive surface functions, it is worthy to comprehend the overall model construction and assessment process of the ML approach. Similar to the cyclic three-concept of the RSM, ML also contains an iterative process to reach a satisfactory estimation result.

ML requires the pre-acquisition of a good amount of experimental data (i.e., the number of samples should be more than the number of coefficients to be estimated. The more samples, the better). A model is constructed using part of the data (training set) and assessed on the remaining data (test set). This technique will help to eliminate the risk of the overfitting problem (the model predicts superior on the current data, yet inferior on unseen data), and therefore to ensure the reliability of the constructed model even when new experimental data becoming available.

In addition to the train-test split technique, ML also employs the cross-validation technique to further assist in preventing the overfitting issue and simultaneously help to select the hyper-parameters (parameters that requires pre-definition by the researcher). When deploying the cross-validation technique, the training set is divided into multiple folds of smaller sets. One of the fold is held as the validation set to assess the prediction power of the constructed model and the rest is utilized to construct the model. The process of model building and validation repeats until each sub-fold having been used as a validation set. The final goodness-of-fitness of the model going through the cross-validation technique is computed as the average of each performance values in the loop.

The following part of the section will introduce a few commonly used regression ML models (learning a function mapping from the influence factors to the responsive factor based on available influence-response pair data) as the responsive surface function:

- the advanced linear regression models (the least absolute shrinkage and selection operator model and the generalized linear model).
- the tree-based models (decision trees, random forest and the gradient boosting decision trees).
- a basic type of the Neural Nets, i.e. the multiple layer perceptrons and
- support vector regression.

### **2.2 Linear regression methods**

#### *2.2.1 Traditional RSM method/the ordinary Least Square (OLS)*

RSM favors the low-order polynomial as the postulation of the mathematical function where the coefficients of the polynomial are estimated by finding the optimal solution to minimize the sum of squared error of the observed response values and the predicted response values. In ML term, the traditional RSM approach to approximate the responsive surface function is referred to as the ordinary least squared model (OLS).

Take the first-order polynomial as an example [29], i.e.

$$\hat{\mathbf{y}} = \boldsymbol{\beta}\mathbf{X}, \quad (1)$$

where  $\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_m)$  stands for the  $m$  coefficients to be estimated,  $\hat{\mathbf{y}}$  is the approximated values of the responsive factor and  $\mathbf{X}$  is the matrix of the influence factors. The optimization problem is to find the  $\boldsymbol{\beta}$  that minimizes the sum of the squared error [29], i.e.

$$\arg \min_{\boldsymbol{\beta}} \|\boldsymbol{\beta}\mathbf{X} - \mathbf{y}\|_2^2, \quad (2)$$

where the  $l^p$ -norm of a vector  $\mathbf{u} = (u_1, u_2, \dots, u_N)$  is defined as:  
 $\|\mathbf{u}\|_p = \left( \sum_{i=1}^N |u_i|^p \right)^{1/p}$  [30].

### 2.2.2 The least absolute shrinkage and selection operator model (LASSO)

LASSO is an upgraded version of the OLS as it allows influence factors selection during the coefficients estimation process and generalization (a technique to avoid overfitting) [23]. LASSO is also a linear regression model yet differs from the OLS by adding an extra regularization term to the loss function to realize the two additional functions [31], i.e. the optimization problem then becomes:

$$\arg \min_{\boldsymbol{\beta}} \underbrace{\frac{1}{2n} \|\boldsymbol{\beta}\mathbf{X} - \mathbf{y}\|_2^2}_{\text{loss function}} + \underbrace{\alpha \|\boldsymbol{\beta}\|_1}_{\text{regularization term}}, \quad (3)$$

where  $\alpha$  is a constant of the  $l^1$ -norm regularization term. Other ML models offering similar functions include the bridge model (where the loss function contains an extra  $l^2$ -norm term other than a  $l^1$ -norm term) and the Elastic-Net model (where the loss function contains both the  $l^1$ -norm and  $l^2$ -norm terms).

### 2.2.3 The generalized linear model

Another useful linear ML model is the generalized linear model (GLM) which allows estimating the response factor when the residuals of the responses do not follow a Gaussian distribution, e.g. the response factor is always positive, or constant value changes in the influence factors leads to exponential value varying other than constant varying of the response factor. In this case, GLM can be utilized to approximate the responsive surface function. It elevates the OLS by differing in two aspects, the predicted value of the response factor is linked to an inverse function of the linear combination of influence factors, i.e.  $\hat{\mathbf{y}} = h(\boldsymbol{\beta}\mathbf{X})$  and the residual term in the loss function is replaced by the unit deviance of a reproductive exponential dispersion model (EDM) [32], i.e.

$$\arg \min_{\boldsymbol{\beta}} \frac{1}{2n} \sum_{i=1}^n d(y_i, \hat{y}_i) + \frac{\alpha}{2} \|\boldsymbol{\beta}\|_2, \quad (4)$$

where  $n$  is the number of samples in the training set. Since the loss function also contains a regularization term, the GLM provides feature selection and generalization during model construction as LASSO does.



| Distribution     | Response domain           | Unit deviance $d(y, \hat{y})$                                 |
|------------------|---------------------------|---|
| Gaussian         | $y \in (-\infty, \infty)$ | $(y, \hat{y})^2$  |
| Poisson          | $y \in [0, \infty)$       | $2\left(y \ln \frac{y}{\hat{y}} - y + \hat{y}\right)$         |
| Gamma            | $y \in (0, \infty)$       | $2\left(\ln \frac{\hat{y}}{y} + \frac{\hat{y}}{y} - 1\right)$ |
| Inverse Gaussian | $y \in (0, \infty)$       | $\frac{(y, \hat{y})^2}{y\hat{y}^2}$                           |

**Table 1.**  
*The unit deviance functions for response factors following various distributions.*

Examples of the unit deviance of the EDM are given in **Table 1** [33].

The 1st-order polynomial assumption is made for purpose of simplifying the concept to introduce the above linear ML models. The intersection and quadratic terms of a higher-order polynomial can be easily computed and added to by performing a transformation on the 1st-order polynomial. The linear regression models described above are still suitable to estimate the coefficients of the transformed influence factors to form a higher-order polynomial responsive surface function.

As mentioned, non-linearity may exist between the influence factors and as such, the polynomials will become inadequate to approximate the mapping from the influence factors to the responsive factor. In this case, the non-linear ML models will become distinctive.

2.3 Tree-based methods

2.3.1 Decision trees

Decision trees (DT), instead of the linear regression models described in Section 2.2, is a non-parametric ML model (models defined without coefficients). The problem is to build a model to predict the values of the responsive factor by means of defining a series of decision principles deduced from the training data [29].

A DT model is built in a top-down manner with each split node partitioning the influence factors into a subgroup and the process eventually reaches a value of the responsive factor [24]. The more important the split node, the higher the node in the tree. The common criteria to minimize as to decide the orders for future split nodes are mean squared error, Poisson deviance and the mean absolute error [34]. Assuming to use the Poisson deviance, suppose the data at the node  $m$  is represented by  $Q_m$  with  $N_m$  samples, the loss function at the split node is defined as:

$$H(Q_m) = \frac{1}{N_m} \sum_{y \in Q_m} y \ln \frac{y}{\bar{y}_m} - y + \bar{y}_m, \tag{5}$$

where  $\bar{y}_m$  is the mean of the responsive values at the node  $m$  and the optimization problem is to identify the node  $m$  that minimizes  $H(Q_m)$  [25].

In comparison to the linear regression models, DT can fit non-linear systems due to the nature of how it is constructed. Besides, since the importance of each influence factor is computed during the tree construction process, it also helps to select the most significant influencers as a procedure of the feature selection and thus to improve the prediction accuracy of the resulted model. However, there are hidden drawbacks when implementing the DT [34]:

- Overfitting problem, i.e. an over-complex DT is built, resulting in good prediction in existing data but poor prediction in unseen data.
- Unstable model, i.e. slight variation in the data can lead to a completely different DT.

### 2.3.2 Random forests and gradient boosting decision trees (GBDT)

In order to address the disadvantages of the DT and to estimate a satisfactory model, ensemble techniques are explored. The ensemble technique combines the prediction of multiple base estimators to achieve reduced variance and enhanced robustness over a single meta-estimator. DT is a common model used as a type of base estimator to form the final meta-estimator. The DT used in an ensemble model is usually simple-structured by limiting the maximum depth of the trees or the maximum leaf nodes of the trees and as a consequence to ease the overfitting issue of using a single DT model.

Two types of ensemble trees, the random forests and the GBDT are introduced here.

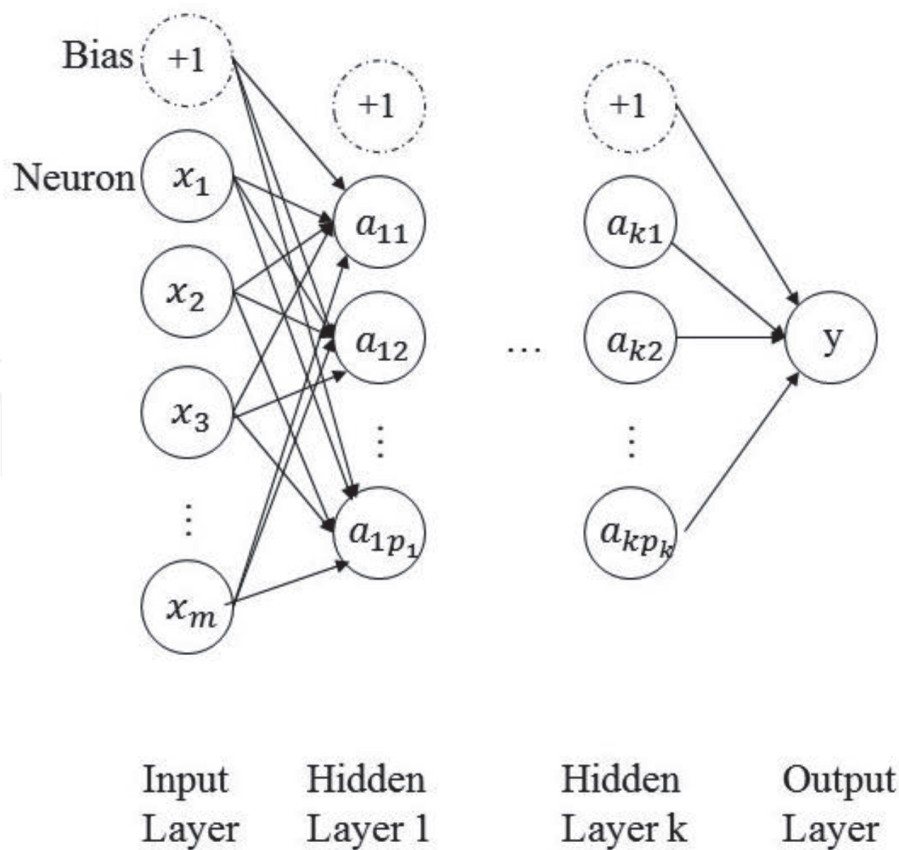
Random forest constructs multiple DT with each DT built from a subset drawn with replacement from the training dataset and uses the averaged prediction of the individual DT as the final prediction for the meta-estimator [29]. GBDT builds a sequence of DT with each preceding DT attempting to eliminate the error of the current sequential DT model and uses a weighted sum of the predictions generated by the sequentially built DT to produce the final prediction [29]. More details regarding the two models can be found in [35, 36].

## 2.4 The neural-nets method

Neural-nets models are a group of models originally inspired by the biological neural networks and are able to learn a complex function mapping from the influence factors to the responsive factor. Neural-nets models have popularized since their development due to their accuracy in predicting without knowing the underlying relationship between the influencers and the responders and therefore massive descendent models have been published recently. In this section, the first generation and the most fundamental neural-net model, the multiple layer perceptrons (MLP) is presented [37].

MLP builds a non-linear function approximation to map the set of influence factors to the responsive factor using the training data. Between the influence factors and the final response factor, there may exist one or multiple non-linear layers, as illustrated in **Figure 1** [29].

Each circle in **Figure 1** is a neuron. The leftmost layer is the input layer that consists of the neurons representing the influence factors. Each neuron in the hidden layer is a weighted linear summation of the neurons in the previous layer followed by a non-linear transformation by applying an activation function. The value of the response factor is given by the neuron in the output layer after receiving the values from the last hidden layer and transforming the values using the linear summation and an appropriate activation function. For the MLP regression model, the activation function in the last step is an identity function, i.e. no activation function is applied in the last step. Similar to the advanced linear models, MLP employs the sum of the squared error loss and an additional  $l^2$ -norm regularization term as the loss function [38], the optimization problem is thus to identify the  $\beta$  that minimizes:



**Figure 1.**  
 Structure of a multiple layer perceptrons (MLP) model.

$$\underset{\beta}{\operatorname{argmin}} \underbrace{\frac{1}{2} \|\beta \mathbf{X} - \mathbf{y}\|_2^2}_{\text{loss function}} + \underbrace{\frac{\alpha}{2} \|\beta\|_2^2}_{\text{regularization term}}, \tag{6}$$

MLP utilizes the stochastic gradient descent (SGD) to update the coefficients based on the gradient of the regularization term and the loss function at each iteration in order to obtain the optimal values of the coefficients [38], i.e.

$$\beta \leftarrow \beta - \gamma \left( \alpha \frac{\partial R(\beta)}{\partial \beta} + \frac{\partial cost}{\partial \beta} \right), \tag{7}$$

where  $\gamma$  is a pre-defined learning rate that controls the step-size in the iteration to reach a local extreme minimum,  $R(\beta)$  is the regularization term and *loss* is the loss function in Eq. (7) respectively.

### 2.5 Support vector regression

Support Vector Regression (SVR) was developed in the 1990s [39], a decade late than the surge of the neutral-nets [40]. Unlike the Neutral-nets, which are result-oriented ‘black-box’ models, the SVR has well-defined underpinned theoretical properties.

Support vectors are the data points from the training set that have a direct determining impact on the optimum location of the decision surface (a hyperplane separating one class of data points from another) [41]. The SVR outstands when there is a limited number of experimental data, in particular, when the number of



samples in the training set is less than the number of influence factors, as the SVR fit a mathematical model by utilizing a subset of the training samples to decide the decision surface. It enhances the linear approximation models by means of fitting non-linear properties in the data as during the model construction process, the influence factors can go through a pre-define non-linear kernel function and as such to realize a non-linear transformation to obtain the response [29].

The procedure to obtain the predicted response values using SVR is similar to the MLP and is illustrated in **Figure 2**. The support vectors are first transformed using and a map  $\varphi$  and then conduct the dot product to evaluate the kernel function

$k(\mathbf{x}, \mathbf{x}_i)$  (for examples, the Gaussian kernel function is given  $k(\mathbf{x}, \mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}}$ ). The values of the kernel functions are then added up using certain weights to achieve the predicted value of the response factor.

Assuming for the 1st-order polynomial approximation (as represented in Eq. (1)), SVR searches for the coefficients by minimizing the inner product of the coefficients, i.e.

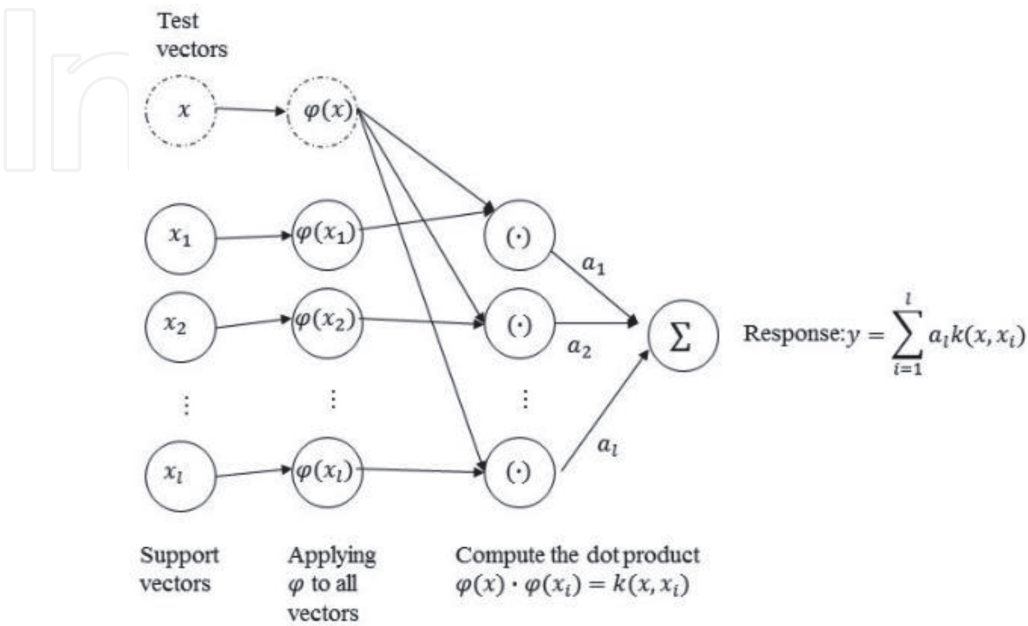
$$\frac{1}{2} \|\beta\|_2 = \langle \beta \bullet \beta \rangle, \quad (8)$$

subjected to the condition of limiting the prediction error into a certain threshold, i.e.  $|\beta\mathbf{X} - \mathbf{y}| < \varepsilon$ , where  $\varepsilon$  is a pre-defined threshold value. However, it is inevitable that not all data points will fall into the threshold and as such, the equation needs to consider the possibility of errors larger than  $\varepsilon$  [41]. Therefore, the equation is completed with an inclusion of the slack variable  $\xi$  as the deviation from the error threshold  $\varepsilon$ . The optimization problem then becomes:

$$\operatorname{argmin}_{\beta} \left( \frac{1}{2} \|\beta\|_2 + C \sum_{i=1}^n \xi_i \right), \quad (9)$$

with constrains  $|y_i - \beta x_i| \leq \varepsilon + \xi_i$  for each  $i = 1, 2, \dots, n$ , and  $\xi_i \geq 0$  [41].

This minimization problem can be resolved by finding the solution of an equivalent Lagrangian Dual problem, i.e. finding the  $\xi$  that maximize:



**Figure 2.**  
The architecture of the support vector regression (SVR) to obtain a prediction.

$$\operatorname{argmax}_{\xi} \left( \sum_{i=1}^n \xi_i - \frac{1}{2} \sum_{i=1}^n \xi_i \xi_j y_i y_j k(\mathbf{x}_i \mathbf{x}_j) \right), \quad (10)$$

subjected to the conditions  $\sum_{i=1}^n \xi_i y_i = 0$  and  $\varepsilon \geq \xi_i \geq 0$ . [41]

### 3. A mechanical engineering case study

The traditional RSM method and the introduced ML models above will be employed to approximate the underlying mathematical model for a set of mechanical engineering data used in [42].

#### 3.1 Description of the experiment data

In the manufacturing processes process, the machine vibration severity level is a critical index to indicate the status of the machine tool and the finish of the cutting material. If a high severity level occurs, the manufacturing process is likely going wrong such as the occurrence of chatter or breakage of a machine tool. However, it is not always loss-effective to have the corresponding devices installed and human technicians in-place to monitor the vibration severity continuously. Instead, a numerical approximation can be explored by collecting machining data, which is easier and less expensive to access.

The experimental dataset contains 56519 samples and 74 variables collected from the sensors installed on the shop floor machine and the central controller computer [42]. To simplify the case to a single response factor problem, the severity variables along different directions measured on various ranges are compacted together using their standardized  $l^2$ -norm values, i.e., the 8 severity columns are first standardized by removing their column mean and scaling to the unit variance. The unified columns are then used to compute the  $l^2$ -norm in the horizontal direction to get the target response variable column. After removing the two time-related columns, the repeated program number column, finally, the processed data provides 63 influence factors and one response factor.

#### 3.2 Implementation of the traditional RSM and the ML methods

A piece of program code is written in Python language and utilizes a package named Scikit-learn [29] to realize the discussed ML models. The main objective here is to demonstrate the advantages of ML in estimating the response surface function to predict the values of the response factor with higher accuracy compared with the traditional RSM polynomial approximation.

First of all, the dataset is standardized to its unit variance form in the same manner as to compute the single severity column in Section 3.1, i.e., using the formula:  $z = \frac{x - \text{mean}(x)}{\text{std}(x)}$ . This is due to that many models to be implemented later require the standardization of the data to avoid the multiple issues (obscuring the conclusion of the statistical significance of the model terms, producing imprecise coefficients or incorrect model structure) caused by the collinearity.

The standardized dataset (containing the 56512 samples data) is divided into a training set and a held-out test set with a 7/3 ratio. The training dataset is further split into three subfolders with each subfolder utilized as a validation set to estimate the prediction power of the model constructed using the remainder two subfolders to adopt the cross-validation technique described in Section 2.1. Employment of the

cross-validation technique will further prevent the over-fitting problem and as such to reassure the robustness of the estimated model and better prediction on unseen data. The training set is used to estimate and select the optimal responsive function and the held-out test set is used to exam the prediction accuracy and the reliability of the fitted RSM function on unseen data. The goodness-of-fitness of the estimated function on the training set and the prediction accuracy of the function on unseen data is measured using three different measuring metrics and the outcomes are shown in the cross-validation results and the held-out test set results sections respectively in **Table 2**.

The three measuring metrics evaluate the model performance from different statistical perspectives to ensure a solid conclusion to be reached. The statistical meanings and the computation equations for the three measuring metrics are given below [43]:

- The explained variance (EV) measures the proportion to which a mathematical model accounts for the variance of a given data set and is computed as

$$EV(y, \hat{y}) = 1 - \frac{\text{var}(y - \hat{y})}{\text{var}(y)}, \tag{11}$$

where *var* stands for the variance. The optimal possible of an EV value is 1. The lower the value, the worse the model performs.

- The mean absolute percentage error (MAPE) measures the percentage difference between the actual and the predicted response factor values and is defined as:

$$MAPE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right|, \tag{12}$$

The best possible MAPE value is 0% when every single predicted value matching the actual one. The greater the score, the worse the model performs.

| Metric name/model type    | Linear models |       |       |       |       |       | Non-linear models |       |       |       |
|---------------------------|---------------|-------|-------|-------|-------|-------|-------------------|-------|-------|-------|
| Model name                | OLS/RSM       |       | LASSO |       | GLM   |       | RF                | GBDT  | MLP   | SVR   |
| Polynomial order          | 1st           | 2nd   | 1st   | 2nd   | 1st   | 2nd   | N/A               |       |       |       |
| Cross-validation Results  |               |       |       |       |       |       |                   |       |       |       |
| EV                        | 0.697         | 0.841 | 0.700 | 0.877 | 0.700 | 0.865 | 0.923             | 0.924 | 0.905 | 0.901 |
| MAPE                      | 31.4%         | 88.4% | 31.4% | 19.7% | 30.2% | 17.8% | 13.5%             | 13.0% | 15.3% | 14.9% |
| RMSE                      | 0.660         | 1.62  | 0.656 | 0.630 | 0.656 | 0.660 | 0.333             | 0.331 | 0.372 | 0.352 |
| Held-out Test Set Results |               |       |       |       |       |       |                   |       |       |       |
| EV                        | 0.699         | −1.08 | 0.699 | 0.878 | 0.699 | 0.865 | 0.925             | 0.926 | 0.908 | 0.900 |
| MAPE                      | 31.4%         | 160%  | 31.5% | 19.7% | 30.3% | 18.0% | 13.2%             | 12.9% | 14.6% | 14.3% |
| RMSE                      | 0.657         | 7.74  | 0.658 | 0.609 | 0.657 | 0.639 | 0.328             | 0.325 | 0.367 | 0.348 |

**Table 2.**  
*Experimental results of applying the traditional RSM method and the ML methods on a set of engineering data.*

- The rooted mean squared error (RMSE) measures the actual differences between the actual value and the predicted value from the model and is defined by:

$$\text{RMSE}(y, \hat{y}) = \left( \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \right)^{1/2} \quad (13)$$

The smaller the value, the better the model performs.

The functions that realize the traditional RSM linear model (i.e. OLS), and the six ML models, including the LASSO, the GLM, the random forests (RF), the GBDT, the MLP and the SVR) in the Scikit-Learn package [29] are implemented and utilized to estimate the underlying mathematical function that maps the 63 influence factors to the target responsive factor for the case study dataset. The technique that generates the 2nd order polynomial terms is also deployed to obtain the polynomial estimation of the traditional RSM function. The goodness-of-fitness and the prediction accuracy of the estimated responsive function using the RSM model or each of the ML models are displayed in the corresponding column labeled with the name of the model in **Table 2**. Particularly, for linear models, both 1st-order and 2nd-order polynomial terms have been attempted.

Furthermore, in order to investigate whether the existing experimental data is enough to achieve a robust and accurate responsive surface function, the learning curve, which determines the cross-validation training and validation accuracy scores under different training sample sizes, is also drawn for each of the models and shown in **Figure 3**. More specifically, using a proportion of the training data to perform the cross-validation technique described in Section 2.1. The mean, the minimum and the maximum of the cross-validation results will be shown on the learning curve for each of the subset used.

### 3.3 Experimental results and discussion

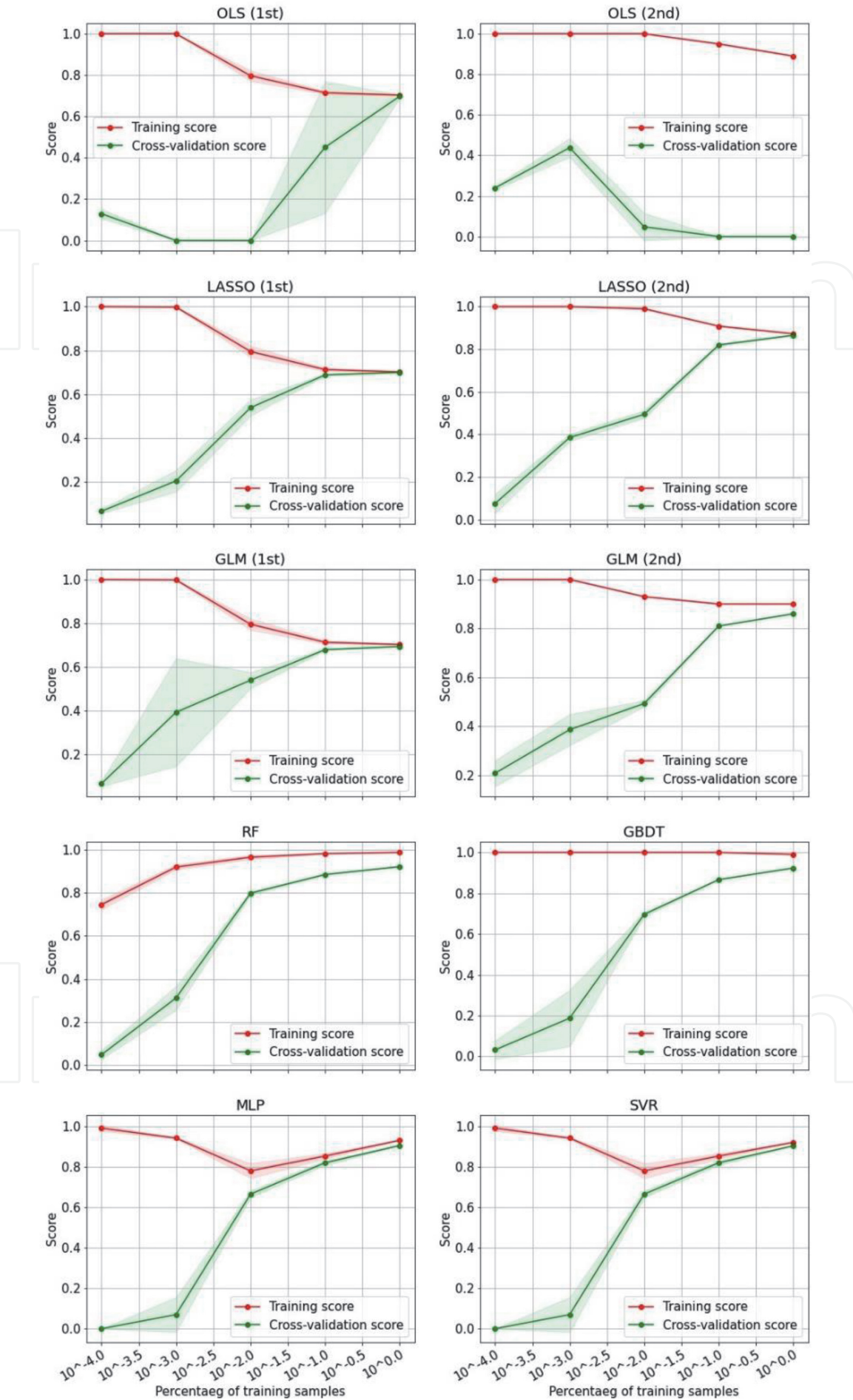
#### 3.3.1 Accuracy of the estimated responsive surface function

The scores in **Table 2** display the cross-validation accuracy and the prediction accuracy on a held-out test set of the RSM and the ML models assessed under three different measuring metrics, the explained variance (EV), the mean absolute percentage error (MAPE) and the rooted mean squared error (RMSE).

Results in **Table 2** has demonstrated that assuming for 1st-order polynomial, the performance of the tradition RSM method, i.e. OLS, is equally mediocre with the other two linear ML models, LASSO and GLM, as all of them produce similar testing results under each of the measuring metrics. This indicates that a 1st-order polynomial assumption may be not enough to include all information between and within the influence factors, which has been validated by the improved model performance for 2nd-order polynomial approximation using a corresponding ML linear model and using the non-linear approximation models.

Under the 2nd-order polynomial postulation, the two linear ML learning models, LASSO and GLM greatly surpass the RSM method as the two models allow influencer selection during model construction and as such to eliminate the influence factors that interferes with estimating precise coefficients. Therefore, more accurate polynomial coefficients are estimated; Besides, both LASSO and GLM perform better under the 2nd-order polynomial assumption than under the 1st-order assumption. The two points imply that some intersection terms of the 2nd-polynomials are redundant and intrusive while the others are necessary to be taken into account. Estimating the





**Figure 3.**  
The learning curves obtained using the traditional RSM method (OLS) and the ML models (LASSO, GLM, RF, GBDT, MLP, and SVR).



coefficients of the intrusive influencers can be a bottleneck for the traditional RSM approximation method as the 2nd-order polynomial approximation using OLS has seen obvious chaos with unreasonably large prediction error.

The table has also demonstrated that the non-linear ML models (RF, GBDT, MLP and SVR) outperform the linear models with or without the intersection and quadratic terms as the measuring metrics have leapt when switching from the linear models to the non-linear ones. Within all the non-linear ML models, the GBDT has exceeded all the others though the performance advantage is relatively small ( $<10\%$ ). In comparison with the traditional RSM method (1st-order polynomial), GBDT has seen a significant improvement (about 50%) on prediction accuracy measured using each of the metrics.

The scores obtained on the held-out test set are almost equal with those achieved through the cross-validation stage for all experimented ML models. The consistency in the metric values reassures the reliability of the constructed model and the prediction accuracy of the model on future unseen data.

### 3.3.2 Size of experimental data

**Figure 3** depicted the learning curve for each of the model described above trained using a proportion of  $10^{-4}$ ,  $10^{-3}$ ,  $10^{-2}$ ,  $10^{-1}$  and  $10^0$  of the original training set (which contains 39563 samples).

Diving into the detail of an individual subplot in **Figure 3**, the red points represent the mean of the training scores while the green ones represent the mean of the validation scores of the cross-validation stage under the training sets with 5 different sizes. The shadowed interval shows the distance between the minimum and maximum of either the training scores (red shadow) or the validation scores (green shadow) under each subset. In cases where the scores are close, the interval can be invisible. The x-axis represents the proportion of the original training set used and the y-axis represents the prediction scores of the trained model. Here, the explained variance is used. The model used to produce the scores is shown on the top of each subplot.

Looking at OLS approximation with the 1st order polynomial terms (the 1st plot on the left), the model is apparently over-fitted when the training set is small. With the increase of the training data size, the validation scores improve but the training scores decrease and the two come to parallel with each other when all training data is used. For the OLS approximation with the 2nd order polynomial terms (the 1st plot on the right), the model is always over-fitted, increasing the training samples does not seem to improve the problem.

For the LASSO approximation with the 1st order polynomial terms (the 2nd plot on the left) and with the 2nd order polynomial terms (the 2nd plot on the right), both of the models improve on predicting the validation set when the overall training set is up-scaled. However, the training scores decrease at the same time.

The same trend applies to the GLM approximations (as shown in the two plots on the 3rd line).

The RF model and the GBDT model have displayed a more satisfying pattern. As shown in the two plots on the 4th line, the validation scores increase or remain as a constant together with the increase of the validation scores when the training samples grow. Even when all training samples are used, the validation score is still lower than the training score. This indicates that there is a chance for a better-fitted model if more experimental data is to be used.

For the MLP model and the SVR model (as shown in two plots on the last line), the training scores are climbing after experienced an inflection point when 0.01 proportion of the training data is used and the validation scores continue to rise

with more data involved. Though the training and the validation scores are neck and neck when the whole training set is used, both of training and the validation scores still get space for improvement.

Considering the general law that the testing scores will not exceed the training scores and the actual scores shown in the plots, we can conclude that all of the RF, GBDT, MLP and SVR models have the potential to train a more accurate responsive surface function if more experimental data becoming available in the future. The other models have already reached their limitations using the current training set and will see little improvement with larger training data.

### 3.3.3 Embedding the simulation technique

A finite element method such as the Monte Carlo simulation can be applied in two ways to complement the RSM study, either to assist in obtaining a better responsive surface function estimation or to achieve the extremums of the existing function.

Instead of collecting more experimental data, synthetic data can be generated using the known knowledge of value intervals, categorical levels or distributions of the influence factors obtained from the existing data. These synthetic data can be populated in pairs of inputs and responses and then be used as a supplement of the current training set aiming to train a more accurate responsive surface function (i.e., to train new RF, GBDT, MLP and SVR models using the up-scaled data). Alternatively, generating the inputs data solely to feed into the obtained mathematical model to attain a corresponding response. Then, picking the pair of influence factors and responsive factor values leading to the global minimum or maximum as the extremums. The simulation technique will not be further discussed here in this book chapter.

## 4. Conclusion

In summary, this book chapter has introduced and discussed

1. two linear ML models (LASSO, GLM) and four non-linear ML models (random forests, GBDT, MLP, and SVR) as alternatives to estimate the responsive surface function.
2. The two linear models use the same optimization function as the traditional RSM method (i.e., OLS in the ML term) to estimate the optimal coefficients of the assumed polynomial yet exceed the OLS by adding an extra regularization term to help to eliminate the redundant and intrusive influencer factors. The improvement can greatly save the efforts on attempting different combinations of influence factors (especially with the higher-order polynomial terms) and solving the optimization function repetitively.
3. The non-linear ML models can pick up the non-linearity across the influence factors that cannot be modeled by the linear models and therefore lead to more precise prediction accuracy.

The advantages of using the ML approached models have been demonstrated by the mechanical engineering case study in Section 3.

1. Results in **Table 2** has shown that all the ML models outperform the traditional RSM polynomial approach. The non-linear models have produced dramatically

improved prediction accuracy. In particular, the GDBT model has shown to exceed the OLS for about 50% on prediction accuracy under each of the measuring metrics.

2. The investigation on the size of the experimental data, i.e. the learning curves in **Figure 3**, has shown that the four non-linear ML models are capable to produce a model with higher accuracy if more training data becoming available.
3. Last yet not least, the simulation technique has been introduced. This technique is essential in either the physical-based or the computer-based experiments to assist in further improving the estimation of the responsive surface function or obtaining the extremums of the current function.

Despite the case study is to predict the vibration severity of the manufacturing machine, the utilization of the ML methods in the RSM can be extended to solve many other engineering problems such as (but not limited) to predict the machine tool life, to estimate the reliability of a structural material or to optimize a bioengineering process where appropriate.

### Author details


Yang Zhang<sup>1\*</sup> and Yue Wu<sup>2</sup>

<sup>1</sup> Data Scientist, London, UK

<sup>2</sup> Mathematical Institute, University of Oxford, Oxford, UK

\*Address all correspondence to: [y.zhang1205@gmail.com](mailto:y.zhang1205@gmail.com)

### IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

## References

- [1] Box GEP, Wilson KB. On the experimental attainment of optimum conditions. *J R Stat Soc Ser B*. 1951;
- [2] Box GEP, Draper NR. A basis for the selection of a response surface design. *J Am Stat Assoc*. 1959;54(287):622–654.
- [3] Box GEP, Hunter JS. Multi-Factor Experimental Designs for Exploring Response Surfaces. *Ann Math Stat* [Internet]. 1957 Mar 1 [cited 2021 Apr 11];28(1):195–241. Available from: <http://projecteuclid.org/euclid.aoms/1177707047>
- [4] Hill WJ, Hunter WG. A Review of Response Surface Methodology: A Literature Survey. 1966;8(4):571–590.
- [5] Myers RH. Response surface methodology - Current status and future directions. *J Qual Technol* [Internet]. 1999 [cited 2021 Apr 11];31(1):30–44. Available from: <https://www.tandfonline.com/doi/abs/10.1080/00224065.1999.11979891>
- [6] Myers RH, Montgomery DC, Geoffrey Vining G, Borror CM, Kowalski SM. Response Surface Methodology: A Retrospective and Literature Survey [Internet]. Vol. 36, *Journal of Quality Technology*. American Society for Quality; 2004 [cited 2021 Apr 11]. p. 53–78. Available from: [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)1084-0699\(2004\)36:1\(53\)>](https://ascelibrary.org/doi/abs/10.1061/(ASCE)1084-0699(2004)36:1(53)>)
- [7] Mead R, Pike DJ. A Biometrics Invited Paper. A Review of Response Surface Methodology from a Biometric Viewpoint. *Biometrics* [Internet]. 1975 Dec 1 [cited 2021 Apr 11];31(4):803. Available from: <https://www.jstor.org/stable/2529809?origin=crossref>
- [8] Box GEP, Draper NR. Response Surfaces, Mixtures, and Ridge Analyses [Internet]. 2nd ed. 2007 [cited 2021 Apr 11]. Available from: <https://www.wiley.com/en-sg/Response+Surfaces%2C+Mixtures%2C+and+Ridge+Analyses%2C+2nd+Edition-p-9780470053577>
- [9] Myers RH, Montgomery DC, Anderson-Cook C. Response Surface Methodology. 4th ed. New Jersey: Wiley Online Library; 2016. 856 p.
- [10] Pike DJ, Box GEP, Draper NR. Empirical Model-Building and Response Surfaces. *J R Stat Soc Ser A (Statistics Soc*. 1988;
- [11] Venter G, Haftka RT, Starnes JH. Construction of response surfaces for design optimization applications. In: 6th Symposium on Multidisciplinary Analysis and Optimization. 1996.
- [12] Box GEP, Tidwell PW. Transformation of the Independent Variables. *Technometrics*. 1962;
- [13] Box GEP, Cox DR. An analysis of transformations. *J R Stat Soc Ser B*. 1964;
- [14] Long T, Wu D, Guo X, Wang GG, Liu L. Efficient adaptive response surface method using intelligent space exploration strategy. *Struct Multidiscip Optim*. 2015;
- [15] Kumar A, Kumar V, Kumar J. Multi-response optimization of process parameters based on response surface methodology for pure titanium using WEDM process. *Int J Adv Manuf Technol*. 2013;
- [16] de Oliveira LG, de Paiva AP, Balestrassi PP, Ferreira JR, da Costa SC, da Silva Campos PH. Response surface methodology for advanced manufacturing technology optimization: Theoretical fundamentals, practical guidelines, and survey literature review. *Int J Adv Manuf Technol*. 2019;



- [17] Mandenius CF, Brundin A. Bioprocess optimization using design-of-experiments methodology. *Biotechnology Progress*. 2008.
- [18] Gilmour SG. Response surface designs for experiments in bioprocessing. *Biometrics*. 2006;
- [19] Edmondson RN. Agricultural response surface experiments based on four-level factorial designs. *Biometrics*. 1991 Dec;47(4):1435.
- [20] Khuri AI. Response Surface Methodology and Its Applications In Agricultural and Food Sciences. *Biometrics Biostat Int J* [Internet]. 2017 Apr 11 [cited 2021 Apr 11];5(5). Available from: <http://medcraveonline.com>
- [21] Mitchell TM. Machine learning and data mining. *Commun ACM*. 1999;
- [22] Alpaydin E. Introduction to Machine Learning. 2014.
- [23] Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;
- [24] Breiman L, Friedman JH, Olshen RA, Stone CJ. Classification and regression trees. Belmont; 2017.
- [25] Hastie T, Tibshirani R, Friedman JH (Jerome H. The elements of statistical learning : data mining, inference, and prediction [Internet]. 2nd ed. New York: Springer; 2009 [cited 2018 Jun 1]. 745 p. Available from: <http://www-stat.stanford.edu/~tibs/ElemStatLearn/>
- [26] Montgomery DC. Design and Analysis of Experiments. 8th Editio. John Wiley & Sons; 2012.
- [27] Gunst RF, Myers RH, Montgomery DC. Response Surface Methodology: Process and Product Optimization Using Designed Experiments. *Technometrics*. 1996;
- [28] What are Taguchi designs. In: *e-Handbook of Statistical Methods* [Internet]. 2012. Available from: <https://www.itl.nist.gov/div898/handbook/pri/section5/pri56.htm>
- [29] Pedregosa, F. Varoquaux, G. Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, et al. Scikit-learn: Machine learning in Python. *J Mach Learn Res* [Internet]. 2011 [cited 2021 Mar 4]; Available from: <https://scikit-learn.org/0.23/>
- [30] Vector Norm [Internet]. Wolfram Mathworld. [cited 2021 Apr 11]. Available from: <https://mathworld.wolfram.com/VectorNorm.html>
- [31] Kim S-J, Koh K, Lustig M, Boyd S, Gorinevsky D. An Interior-Point Method for Large-Scale 1-Regularized Least Squares. *IEEE J Sel Top Signal Process*. 2007;1(4).
- [32] McCullagh P, Nelder J. Generalized Linear Models [Internet]. Chapman and Hall; 1983. Available from: <http://www.utstat.toronto.edu/~brunner/oldclass/2201s11/readings/glmbook.pdf>
- [33] Generalized Linear Regression [Internet]. scikit learn. 2021 [cited 2021 Feb 24]. Available from: [https://scikit-learn.org/stable/modules/linear\\_model.html#generalized-linear-regression](https://scikit-learn.org/stable/modules/linear_model.html#generalized-linear-regression)
- [34] Hastie T, Tibshirani R, Friedman J. Elements of Statistical Learning. 2nd ed. Springer; 2009.
- [35] Breiman LEO. Random Forests. *Mach Learn* [Internet]. 2001;5–32. Available from: <https://link.springer.com/content/pdf/10.1023/A:1010933404324.pdf>
- [36] Chen T, He T, Michael B, Vadim K, Yuan T, Hyunsu C, et al. Extreme Gradient Boosting [Internet]. CRAN.



[cited 2020 Sep 2]. Available from:  
<https://cran.r-project.org/web/packages/xgboost/xgboost.pdf>

[37] Rumelhart E. David, Geoffrey HE, J. WR. Learning representations by back-propagating errors. Lett to Nat [Internet]. 1986;323(9). Available from: [https://www.iro.umontreal.ca/~pift6266/A06/refs/backprop\\_old.pdf](https://www.iro.umontreal.ca/~pift6266/A06/refs/backprop_old.pdf)

[38] Kingma DP, Lei Ba J. Adam: a method for stochastic optimization. In: The 3rd International Conference on Learning Representations [Internet]. San Diego; 2015. p. 1–15. Available from: <https://arxiv.org/pdf/1412.6980v8.pdf>

[39] Boser BE, Guyon IM, Vapnik VN. A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on Computational learning theory [Internet]. 1992. Available from: <https://dl.acm.org/doi/10.1145/130385.130401>

[40] Berwick R. An Idiot's guide to Support vector machines [Internet]. p. 1–28. Available from: <http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf>

[41] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput. 2004;

[42] Zhang Y, Beudaert X, Argandona J, Ratchev S, Munoa J. A CPPS based on GBDT for predicting failure events in milling. Int J Adv Manuf Technol. 2020;

[43] Hyndman RJ, Athanasopoulos G. Forecasting: Principles and Practice [Internet]. 2018 [cited 2018 Dec 6]. Available from: <https://otexts.org/fpp2/>